

# Mise en place d'un site wordpress couplé à un node-exporter

---

WordPress est un outil de blogging gratuit et open source et un système de gestion de contenu basé sur PHP et MySQL, qui s'exécute sur un service d'hébergement Web.

Nous allons utiliser docker-compose pour déployer un site wordpress prêt à l'emploi en le configurant via un fichier docker-compose.yml.

Le fichier docker-compose.yml se définit comme suit:

```
version: '3.3'

services:
  web:
    image: wordpress:latest
    container_name: wordpress-somily
    volumes:
      - ./wordpress/volumes:/var/www/html
    environment:
      - WORDPRESS_DB_HOST=141.94.6.207:3306
      - WORDPRESS_DB_USER=wordpress
      - WORDPRESS_DB_PASSWORD=wordpress
      - WORDPRESS_DB_NAME=wordpress
    labels:
      - "prometheus_job=wordpress"
      - "prometheus.path=/metrics"
    ports:
      - 80:80
      - 443:443
```

Ce fichier docker-compose définit un service appelé "web" qui exécute la dernière version de WordPress avec la commande : `wordpress:latest`. Le service une fois lancé sera nommé "wordpress-somily".

## Définition des configurations et des options

**Image :** L'image de WordPress utilisée est la dernière version disponible de WordPress depuis Docker Hub.

**Volumes :** Les fichiers d'installation de WordPress sont stockés dans le répertoire `./wordpress/volumes/` sur la machine hôte, et seront stockés dans le répertoire `/var/www/html` du conteneur **wordpress-somily**. Cela permet à toutes les modifications apportées dans le conteneur WordPress d'être persistantes sur la machine hôte.

### Variables d'environnement :

- `WORDPRESS_DB_HOST=141.94.6.207:3306` : Définition de l'hôte de la base données du site wordpress.
- `WORDPRESS_DB_USER=wordpress` : Définition de l'utilisateur sur la base de données du site wordpress.

- `WORDPRESS_DB_PASSWORD=wordpress` : Définition du mot de passe sur la base de données du site wordpress.
- `WORDPRESS_DB_NAME=wordpress` : Nom de la base de données du site wordpress.

Ces variables d'environnement permettent au conteneur WordPress de se connecter au conteneur MySQL de la base de données qui s'exécute sur le même réseau.

**Labels :** Deux labels, ou étiquettes en français sont ajoutées au conteneur, qui peuvent être utilisées par un système de surveillance comme Prometheus pour suivre et extraire des métriques à partir du conteneur.

L'étiquette `prometheus_job=wordpress` identifie ce conteneur en tant que service WordPress, et l'étiquette `"prometheus.path=/metrics"` spécifie le chemin où le conteneur expose ses métriques.

**Ports :** Le conteneur WordPress est configuré pour exposer les ports 80 et 443, ce qui permet au trafic externe d'atteindre le site WordPress. Le **port 80** est le port HTTP standard, tandis que le **port 443** est le port HTTPS standard.

Dans ce même fichier docker-compose.yml qui exécute wordpress, on peut exécuter un service node-exporter à la suite définit comme suit:

```
node-exporter:
  image: prom/node-exporter:latest
  container_name: node-exporter
  restart: unless-stopped
  volumes:
    - /proc:/host/proc:ro
    - /sys:/host/sys:ro
    - /:/rootfs:ro
  command:
    - '--path.procfs=/host/proc'
    - '--path.rootfs=/rootfs'
    - '--path.sysfs=/host/sys'
    - '--collector.filesystem.mount-points-exclude=^/(sys|proc|dev|host|etc)($$|/)'
  ports:
    - 9100:9100
```

Le node-exporter collecte les métriques du système d'exploitation et les expose pour être collectées par un outil de surveillance tel que Prometheus.

L'image utilisée pour le node-exporter est la dernière version disponible de **prom/node-exporter** depuis Docker Hub.

Dans la partie volume, on permet au node-exporter d'accéder aux informations système. Les répertoires `/proc`, `/sys` et `/` sont montés en lecture seule (`/proc:/host/proc:ro`, `/sys:/host/sys:ro`, `/:/rootfs:ro`).

Des commandes sont utilisées pour configurer le chemin d'accès aux informations du système dans le node-exporter.

Les commandes `--path.procfs=/host/proc`, `--path.rootfs=/rootfs`, `--path.sysfs=/host/sys` spécifient les chemins pour accéder aux informations du système d'exploitation. La commande `--collector.filesystem.mount-points-exclude=^/(sys|proc|dev|host|etc)($$|/)` exclut les points de montage du système de fichiers qui ne doivent pas être surveillés.

Le node-exporter est configuré pour exposer le **port 9100**. Ce port permet à un outil de surveillance, tel que Prometheus, de collecter les métriques exposées par le node-exporter.

Avec ce service maintenant configuré, les données du conteneur wordpress seront exposées afin d'être collecté par un outil externe tel que Prometheus.